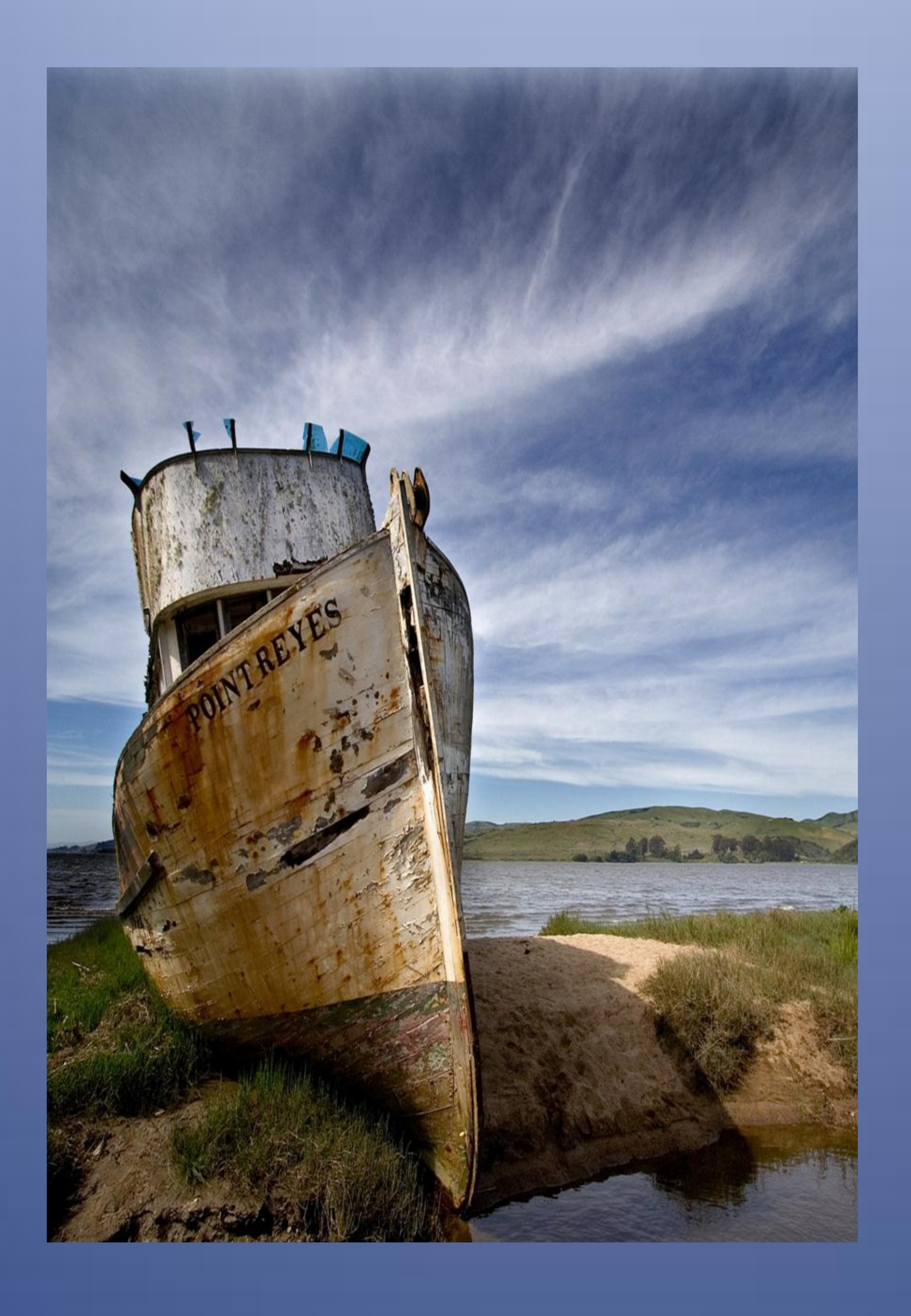


C++ Examples

Solved C++ Examples

In this book you will find many solved examples in C++ programming language Including the advanced object oriented programming style examples and so many notes about programming in C++.

With all the best, By: Mohammed Gohdar

Please Read This

May be you will find some mistakes in this book this shouldn't surprise you, because this book is the work of single person, and me (the author) is not responsible about any mistakes that will cause you to get wrong information.

To help others and make this book more helpful to you and your Friends Please If you found any mistakes contact me:

Michael.nur@facebook.com

By: Mohammed Gohdar Mohammed

From: University Of Duhok – Computer Science

Publishing year: 2012 – First Edition

Thank you for your attention. ☺

بسم الله الرحمن الرحيم

((يرفع الله الذين ءامنوا والذين أوتوا العلم درجات))

صدق الله العضيم

الشيئان اللذان ليس لهما حدود، الكون وغباء الإنسان، مع أني لست متأكدا بخصوص الكون (البرت انشتاين)

حارب عدوك بالسلاح الذي يخشاه ، لا بالسلاح الذي تخشاه(غاندى)

للعظيم قلبان : قلب يتألم ، وقلب يتأمل (جبران)

انظر إلى الإنسان كما هو لا كما كان (مثل صينى)

```
//write a program to output:- 12345678910
#include<iostream.h>
main()
{
        int a ;
        for(a=1;a<=10;a++)
            cout<<a;
            cout<<endl;
}
************************************************************
//write a program to convert the seconds to minutes and hours like 00:00:00 :-
#include<iostream.h>
main()
{
        int second,hours=3600,minutes=60;
        cout<<"Enter the seconds ";
        cin>> second;
        hours=second/hours;
        minutes=second/minutes;
        second=second%60;
        cout<<hours<<":"<<minutes%60<<":"<<second<<endl;
}
************************************************************
```

//write a program to input 3 integers and determine which of them is biggest:-

```
#include<iostream.h>
main()
{
        int a,b,c;
        cout<<"enter a , b and c one after another"<<endl;
        cin>>a>>b>>c;
        if ((a>b)&&(a>c)) //take care of (( )) important.
                cout<<"first number is bigger="<<a<<endl;
        if ((b>a)&&(b>c))
                cout<<"the second number is bigger="<<b<<endl;
        if ((c>a)&&(c>b))
                        cout<<"the last number is bigger="<<c<<endl;
                if((a==c)&&(c==b))
                        cout<<"all the numbers are equal"<<endl;
}
```

****

//write a program to enter a letter and determine the letter is small or capital:-

```
#include <iostream.h>
main()
{
  char letter;
  cout<<"Enter a letter:";
  cin >> letter;
  if(letter >= 'A')
  if(letter <= 'Z')
            cout<<"You entered a capital letter.";
  else
            cout<<"You entered a small letter.";
}
```

****

//Just an Example.

```
#include<iostream.h>
main()
{
        int a=3,b=2,c=a*b;
        b++;
        a=b++;
        b++;
        cout<<a<<endl<<b<<endl<<c<<endl;
}
```

=3=a

=5=b

=6=c

//You try This one

```
#include<iostream.h>
main()
{
        int d,a,A;
        d=A==a;
        cout<<d<<endl;
}
```

=1 because that's true two intiger are equal

//You try this one

```
# include<iostream.h>
main()
{
        char d,a,A;
        d=A==a;
        cout<<d<<endl;
}
```

=:)

why?? i don't know.

****

//write a program to calculate the value of X where x=(3+4ab)/2c :-

```
#include<iostream.h>
main()
{
        float a,b,c,x;
        cout<<"enter a , b and c one after another"<<endl;
        cin>>a>>b>>c;
        x=(3+4*a*b)/(2*c);
        cout<<"x="<<x<<endl;
}
```

****

T.B/ >> mean input, << mean output

T.B/ take care!! of <= that's true, but =< that's Error

>= that's true, but => that's Error

T.B/int=1,2,3,4... , char=a,b,A,T,@,#,$....float= 1.2, 3/2, 0.0 ....

T.B/ a%2 is equal to a%2!=0

T.B/ a+=1 is equal to a=a+1

T.B/ a*=a+b is equal to a=a*a+a*b or equal to a=a(a+b)

T.B/ to output the " in black board you should write it like \" this

T.B/ cout<<"\\ it will output like \ ..But cout<<"// the output is //

T.B/ you should know this (==)equal sign used with "IF"

T.B/ Some time we use "do while" instead of "For" that if we didn't know the number of loops.

****

//write a program to know positive, negative, odd and even numbers

```
#include<iostream.h>
main()
{
        int a;
        cin>>a;
                if(a>=0)
                cout<<"a is positive number\n";
              if(a%2==0)
```

```
      cout<<" a is even \n";
    if(a%2!=0)
     cout<<"a is odd  number"<<endl;
    if(a<0)
     cout<<"a is a negative number\n";
}
```

****

H.W//write a program to know the positive numbers "and" that divisible by 4

```
//Calculator program
#include<iostream.h>
main()
{
    float a,b,c;
    char sign;
        cin>>a;
        cin>> sign;
        cin>>b;
        switch (sign)
        {
        case '+': cout<<"="<<(c=a+b) <<endl ; break;
        case '/': cout<<"="<<(c=a/b) <<endl ; break;
        case '-': cout<<"="<<(c=a-b) <<endl ; break;
        case '*': cout<<"="<<(c=a*b) <<endl ; break;
        } }
```

****

//write a program to calculate the SUM of positive integers and the SUM of negative integers for 6 integers.

```
#include <iostream.h>

main()

{

	int b=0,x=0,mark=0,z=0;

	for(x=1;x<=6;++x)

	{

		cin>>mark;

		if(mark>=0)

			z=mark+z;

		else

			b=mark+b;

	}

	cout<<"sum of positives="<<z<<endl;

	cout<<"sum of negatives="<<b<<endl;

}
```

****

//write a program to compute the sum of even numbers from some numbers until "0" by using DO/WHILE loop:-

```
#include<iostream.h>

int main()

{

    int a,b=0;

    do

    {    cin>>a;

         if(a%2==0)

         b=b+a;

      } while(a!=0);

         cout<<b;

}
```

****

```
//write a program using DO/WHILE to compute factorial:-
#include<iostream.h>
main()
{       int a,b,c=1;cin>>a;
          do {     c=c*a;  a–;
               }while(a!=1);
             cout<<c<<endl;
}
```

****

```
//Write a program to find  N! .
#include <iostream.h>
main()
{
          int a,b,c=1;
          cin>>b;
          for(a=1;a<=b;a++)
                    c=c*a;
          cout<<c<<endl;
}
```

****

```
//write a program to count the sum of an array:-
#include <iostream.h>
main()
{
          int x,z[5],p=0;
          for(x=0;x<5;x++)
                    {  cout<<endl<<"enter z["<<x<<"]=";
                    cin>>z[x];
                    p+=z[x];   }
          cout<<"sum of array ="<<p<<endl;
          cout<<endl;
}
```

****

//write a program to copy array B to D ,B should be chosen by the user:-

```
#include<iostream.h>
main()
{
        int a,b,c[5], d[5];
        for(a=0;a<5;a++)
        {
                cout<<"c["<<a<<"]=";
                cin>>c[a];
                d[a]=c[a];
          }
                  for(a=0;a<5;a++)
                  cout<<"d["<<a<<"]="<<d[a]<<endl;
}
```

****

//write a program to find the maximum and minimum value from an array:-

```
#include <iostream.h>
void main()
{
        int x,z[5];
        for(x=0;x<5;x++)
        {
                cout<<endl<<"enter z["<<x<<"]=";
                cin>>z[x];
        }
        int d=z[0],c=z[0];
                for(x=0;x<5;x++)
                {
                if(d<=z[x]) d=z[x];
                else d=d;
                if(c<=z[x])  c=c;
                else c=z[x];
                }
```

```
        cout<<"Maximum number ="<<d<<endl;
        cout<<"Minimum number ="<<c<<endl;
}
```

****

H.W//write a program to output the smallest number from an array:-

****

```
//important one :-)
#include <iostream.h>
main()
{
        int a=0;
        while(a++<10)
                ++a;
        cout<<a<<endl;
}
a=11
```

****

```
//if you are using numbers for switch cases:-
#include <iostream.h>
main()
{
        int a;
        cin>>a;
        switch(a)
        {
        case 1:cout<<"Hello1\n";break;
        case 2:cout<<"Hello2\n";break;
        }
```

****

```
//if you are using characters for switch cases:-

#include <iostream.h>
main()
{
        char a;
        cin>>a;
        switch(a)
        {
        case '1':cout<<"hello1\n";break;
        case '2':cout<<"hello2\n";break;
        }
}
```

****

*And also you should know that we can't never use (float) for switch cases :).

**//write a program to output the following result (1 2 3 4 2 4 6 8 3 6 9 1 2) by using for loop...

```
#include <iostream.h>
main()
{
        int a,b;
        for(a=1;a<=3;a++)
        for(b=1;b<=4;b++) //the value of a will not change until b's complete
         cout<<a*b<<endl;
}
```

****

//write a barnamj to output:-

```
#
##
###

#include<iostream.h>
main()
{
        int a,b;
        for(a=1;a<4;a++)
        {
                for(b=1;b<=a;b++)
                cout<<"#";
                cout<<endl;
        } }
```

****

//write a program to output:-

```
####
###
##
#
#include<iostream.h>
main()
{
        int a,b;
        for(a=1;a<5;a++)
        {
                for(b=4;b>=a;b--)
                 cout<<"#";
                cout<<endl;
        }
}
```

****

//write a program to output:-

```
####
 ###
  ##
   #
```

```
#include <iostream.h>
main()
{
        int a,b;
        for(a=1;a<=4;a++)
        {
            for(b=1;b<=4;b++)
          if(a>b)
          cout<<" ";
          else  cout<<"#";
          cout<<endl;
          }
}
```

****

//write a program to output:-

```
#
##
###
####
###
##
#
```

```
#include <iostream.h>
main()
{
        int a,b,c;
```

```
        for(a=1;a<=3;a++)
        {
        for(b=1;b<=a;b++)  //take care don't say I can use 3 instead of "a",no you can't
                cout<<"#";
                cout<<endl;
        }
        for(a=1;a<=4;a++)
        {       for(b=4;b>=a;b--)
                cout<<"#";
                cout<<endl;
        }
}
```

****

//write a program to output:-

```
   #
  ###
 #####
#######
```

```
#include<iostream.h>
main()
{
        int a,b,c;
        for(a=4;a>=0;a--)
        {
                for(b=1;b<=a;b++)
                        cout<<" ";
                for(c=0;c<7-(2*a);c++)
                        cout<<"#";
                cout<<endl;
        } }
```

 * understand? No? okay there is another way my friend...see next example...

****

```
//write a program to output:-
    #
   ###
  #####
 #######

#include <iostream.h>
main()
{
        int a,b,c,d=1;
        for(a=4;a>=1;a--)
        {
                  for(b=1;b<a;b++)
                cout<<" ";
                  for(c=1;c<=d;c++)
                cout<<"*";
                cout<<endl;
                d+=2;
        }
}
```

****

```
//Write a program to output:
    A
   BCD
  EFGHI
jKLMNOP
#include <iostream.h>
main()
{
        int a,b,c,d=1,asci=65;

        for(a=4;a>=1;a--)
        {
```

```
                    for(b=1;b<a;b++)
                   cout<<" ";
                    for(c=1;c<=d;c++)
                   cout<<char (asci++);
                   cout<<endl;
                   d+=2;
         }
}
```

****

```
//write a program to output:-
#####
 ###
  #
  *
 ***
*****

#include <iostream.h>
void main()
{
         int a,b,c,d=0,g=5;

         for(a=1;a<4;a++)
         {          for(b=1;b<a;b++)
                   cout<<" ";
                   for(c=1;c<=g;c++)
                   cout<<"#";
                   g-=2;
                   cout<<endl;  }

         for(a=3;a>0;a–)
         {          for(b=1;b<a;b++)
                    cout<<" ";
```

```
                    for(c=0;c<=d;c++)
                    cout<<"*";
                    d+=2;
                    cout<<endl; }
}
```

****

//write a program to output:-

```
##
####
######
```

```
#include<iostream.h>
main()
{
        int a,b;
        for(a=2;a<=6;a+=2)
        {
                for(b=1;b<=a;b++)
                cout<<"#";
                cout<<endl;
        }
}
```

****

//write a program to output:-

```
   ######
  ######
 ######
######
```

```
#include<iostream.h>
main()
{
        int a,b,c;
```

```
        for(a=4;a>=0;a--)
        {
                for(b=0;b<=a;b++)
                cout<<" ";
                for(c=0;c<7;c++)
                 cout<<"#";cout<<endl;
        }
}
```

****

//write a program to output:-

```
    #
   #
  #
 #
#
```

```
#include<iostream.h>
main()
{
        int a,b;
        for(a=4;a>=0;a--)
        {
            for(b=0;b<=4;b++)
            if(a==b)
            cout<<"#";
            else
            cout<<" ";
            cout<<endl;
        }
}
```

****

//write a program to output:-

```
 ###
# ##
## #
###
```

```
#include <iostream.h>
main()
{
        int a,b;
        for(a=1;a<=4;a++)
        {
             for(b=1;b<=4;b++)
           if(a==b)
           cout<<" ";
           else    cout<<"#";
           cout<<endl;
          }
}
```

****

//write a program to output:-

```
#
 #
  #
   #
  #
 #
#
```

```
#include <iostream.h>
main()
{
        int a,b,c;
```

```
for(a=1;a<=3;a++)
{ for(b=1;b<=a;b++)
if(a>b)
cout<<" ";
cout<<"#"; //as you see i didn't used "else" thats mean we haven't use it
cout<<endl;
}
for(a=1;a<=4;a++)
{         for(b=4;b>=a;b--)
            if(a<b)
          cout<<" ";
          cout<<"#";
          cout<<endl;
}
}
```

****

```
//Another professional way
#include <iostream.h>
void main()
{
          int a,b;
          for(a=1;a<=7;a++)
          {
                    for(b=1;b<=4;b++)
                    if((a+b==(2*a))||(a+b==8))
                    cout<<"*";
                    else cout<<" ";
                    cout<<endl;
             }
}
```

****

//write a program to output:- **IMPORTANT!!!**

```
#########
#       #
#       #
#########
```

```
#include<iostream.h>
int main()
{
    for(int i=1;i<5;i++)
   {
     for(int j=1;j<10;j++)
     if(j==1 || j==9 || i==1 ||i==4)
     cout<<"*";
     else cout<<" ";
     cout<<endl;
   }
}
```

****

H.W//write a program to output:- it's really simple you can do that :) gooo a head

```
#########
#       #          ***********
    #                M     M
#       #               J
#########     AND  ***********     scary? ;)
```

****

//write a program to output:-

```
#####
*####*
*####*
*####*
```

```
#include<iostream.h>
main()
```

```
{
        int a,b;
        for(a=1;a<5;a++)
        {
              for(b=1;b<=5;b++)
            if(a==1||b==2||b==3||b==4)
            cout<<"#";
            else cout<<"*";
            cout<<endl;
        }
}
```

****

//write a program to compute the sum of integers that divisible by 4 from 6 integers:-

```
#include<iostream.h>
void main()
{
        int a,b,c=0;
        for(a=1;a<=6;a++)
        {
          cin>>b;
          if(b%4==0)
          c+=b;
        }
        cout<<"c="<<c<<endl;
}
```

****

```
//write a program to find the avarage of 3 student from 3 lecture:-
#include<iostream.h>
void main()
{
        float  mark=0, total=0 ,avarage=0;
        for(float a=1;a<=3;a++)
        {
                total=0;
                        for(float b=1;b<=3;b++)
                        {
                                cin>>mark;
                                total+=mark;
                        }
                        avarage=total/3;
                        cout<<"avarage="<<avarage<<endl;
        }
}
```

****

```
//write a while loop to output the:- 1 3 12 60 360
#include <iostream.h>
main()
{
        int a=1,p=1;
        while(p<360)
        {
                p=p*a;
                a+=1;
                if(a==2)
                a++;
          cout<<p;
        }cout<<endl;
}
```

****

//write a program to output (using nested loop):-   **IMPORTANT!**

2 5 8

5 8

8

```
#include <iostream.h>
main()
{
        int A,b;
        for(A=2;A<=8;A+=3)
        {
                for(b=A ;b<=8 ;b+=3) // take care!!
                    cout<<b;
                    cout<<endl;
        }
}
```

****

//write a program to compute the sum of the integers that divisible by 4   from 10 different integers:-

```
#include <iostream.h>
main()
{
        int a,b,c=0;
        for(a=1;a<=10;a++)
        {
          cin>>b;
          if(b%4==0)
            c+=b;
        }
            cout<<"c="<<c<<endl;  }
```

There we used block because we need more than one integer "B" to compute the sum of integers

Because this program without block mean that we have only one value

Our value will change 10 times and it will take the last value... nothing else. :-/

//write a program to compute the sum of the integers that divisible by 4

From 10 different integers (using DO/WHILE loop) :-

```
#include <iostream.h>
main()
{
        int a=1,b,c=0;
        do
        {
              cin>>b;
               a++;
               if(b%4==0)
               c+=b;
        }while(a<=10);
          cout<<"c="<<c<<endl;
}
```

****

//write a program used nested loop to calculate "X" ,where:-

X= 1!+ 3!+ 5!+ 7!

```
#include<iostream.h>
void main()
{
        int a,b,c=1,d=0;
        for(a=1;a<=7;a+=2)
        {
                    c=1;
                for(b=1;b<=a;b++)
                c*=b;
                cout<<"factorial " <<a<<"= "<<c<<endl;
                d+=c;
        }  cout<<"Sum ="<<d<<endl;
}
```

****

//write a program to output using NESTED loop:-

54321

5432

543

54

5

```
#include <iostream.h>
main()
{
        int a,b;
        for(a=1;a<=5;a++)
        {
                for(b=5;b>=a;b--)
                    cout<<b;
                    cout<<endl;
        }
}
```

....

//write a program by using NESTED loop to output:-

1 2 3 4

2 3 4 5

3 4 5 6

```
#include<iostream.h>
main()
{
        int a,b,c=4;
        for(a=1;a<=3;a++)
        {
           for(b=a;b<=c;b++)
           cout<<b;cout<<endl;  c++;}
}
```

****

//write a program by using NESTED loop to output:-

5 4 3 2 1

6 5 4 3 2

7 6 5 4 3

8 7 6 5 4

9 8 7 6 5

```
#include<iostream.h>
main()
{
        int a,b,c=1;
        for(a=5;a<=9;a++)
        {
                for(b=a;b>=c;b--)
                cout<<b;cout<<endl; c++;
          }
}
```

****

//write a program to output:- IMPORTANT

1

23

345

4567

56789

```
#include <iostream.h>
main()
{
        int a,b,c=1;
        for(a=1;a<=5;a++)
        {
              for(b=a;b<=c;b++)
            cout<<b;
            cout<<endl;  c+=2;    } }
```

****

H.W// write a program to output the following numbers by using (nested FOR loop):-

123345456756789 ;)

****

//write a barnamej using function…that takes three integers as arguments(hours,minutes,seconds) and return the number of seconds:-

```
#include<iostream.h>
int time(int x,int y,int z)
{
        x=x*3600;
        y=y*60;
        z=z+x+y;
        return z;
}
void main()
{
        int x,y,z;
        int seconds;
        cin>>x>>y>>z;
        seconds=time(x,y,z);
        cout<<"seconds= "<<seconds<<endl;
}
```

****

//write a FOR loop to print the following integers:- 1 2 3 5 8 13 21 34 55

```
#include<iostream.h>
void main()
{
        int a=1,b=0,c;
        for(c=1;c<55;c++)
        {
           c=a+b;
           b=a;
```

```
                a=c;cout<<" "<<c;
              } cout<<endl;
}
```

****

//write a program to calculate the value of "Z", where: Z= x^n/n!

```
#include<iostream.h>
main()
{
        float a,b,c=1,x,n,y=1;
        cin>>x>>n;
        for(a=1;a<=n;a++)
                c=c*x;
        for(b=1;b<=n;b++)
                y=y*b;
        cout<<"n!="<<y<<endl;
        cout<<"x="<<c<<endl;
        cout<<"Z="<<c/y<<endl;
}
```

....

//write a barnamj to compute "Z" ,whre

Z=x/1! + x^2/2! + x^3/3!

```
#include<iostream.h>
main()
{
        float i,p,n=1,x,a,f=1,d=1,b=1,g=0,z;
        cin>>x;
        for(i=1;i<=3;i++)
        {
                for(p=1;p<=i;p++)
                        n=n*x;
                for(;d<=b;d++)
```

```
                f=f*d;
            z=n/f;
            g=g+z;
            b++;
            d=1;
            n=1;
            f=1;
        }
        cout<<g<<endl; }
```

****

```
//An example about structer.
#include<iostream.h>
struct student
{
        char name[80];
        int  age;
        struct birth_date
        { int year;
          int month;
          int day;
        }ddd;
};
void main()
{
        student stud;
        cin>>stud.name;
        cin>>stud.age;
        cin>>stud.ddd.year;
        cin>>stud.ddd.month;
        cin>>stud.ddd.day;    }
```

****

//An Example shows you the similarity between arrays and pointers.

```
#include<iostream.h>

void main()

{

int arr[3];

arr[0]=4;

arr[1]=5;

arr[2]=6;

int *p;

p=&arr[0];

cout<<"p= "<<*(p+1)<<endl;

}
```

//p itself is a pointer. But i will need a * sign if I used as an array…like *(p+1)

//Write a member function in your class that insert values to private data members and displays it.

```
#include <iostream.h>

class time{

public:

time(); //Constructor

void display(){ cout<<hour<<endl<<mint<<endl;};

void settime (int a,int b){   hour=a; mint=b;   };

private:

int hour;

int mint;

};

void main()

{

time obj1();

obj1.time2(5,6);  //there we inserted values

}
```

//Calling a private member Function.

```
#include <iostream.h>

class time{

public:

time(int a,int b){hour=a;mint=b; time2(); } //There we called a private member function By a member function in public part.

time3(int a,int b){hour=a;mint=b;}

private:

time();

int hour;

int mint;

void time2(){ cout<<hour<<endl<<mint<<endl;};

};

main()

{  time obj1(5,8);  }
```

//use a destructor and define a member function outside of the class body.

```
#include <iostream.h>

class time

{

public:

time(int a ,int b){ hour=a; mint=b; } ;

void settime(int,int) ;

void printer() ;

~time(){cout<<"The constructor is DESTROYIED"<<endl;}

private:

int hour,mint;

};

void time::printer() // a member function that is defined outside of the class body by using name resolution operator(::) .

{cout<<"hour= "<<hour<<endl<<"mint= "<<mint<<endl; };

void main()

{  time obj1(5,7); obj1.printer();   }
```

//You Try this Example

```
#include <iostream.h>
func(int a) {      a++;     }
main()
{int a,b;
cin>>a;
//        int c=func(a);  see the difference
int c=a;
cout<<c<<endl; }
```

// using a constructor and destructor.

```
#include<iostream.h>
const int SIZE=100;
class stack {
int stck[SIZE];
int tos;
public:
stack( ); // constructor
~stack( ); //destructor
};
// stack.s constructor function
stack::stack( )
{
tos=0;
cout<<"stack Initialized"<<"\n";
}
//constructor function's destructor
stack::~stack( ){cout<<"Stack is destroyed"<<endl;}
void main()
{ stack obj1;  }
```

```
//Just an example.
#include<iostream.h>
class X {
int a;  //By defult is defined in Private part.
public:
X(int j) {a= j;} // a constructor with one argument.
int geta( ) {return a; } };
int main( )
{
X ob = 99; //is the same to  X ob(99);
cout<<ob.geta( ); // outputs 99
return 0;
}
```

```
//Constructor and distractor Defined outside of the class body.
#include<iostream.h>
class myclass {
public:
int who;
myclass(int id); //default constructor
~myclass( );     //destructor
} glob_ob1(1), glob_ob2(2);
myclass::myclass(int id) //Constructor with one argument
{
cout<<"Initializing "<<id<<"\n";
who = id;
}
myclass::~myclass( ) {   cout<<"Destructing "<<who<<"\n";}
int main( )
{
myclass  local_ob1(3);
cout <<"this will not be first line displayed.\n";
myclass local_ob2(4);
```

```
return 0;

}
```

T.B//we can't call the constant objects by the member functions

----

```
//an Example for static Data member

#include<iostream.h>

class shared {

static int a; // Static data member must be inside class followed by static key word, and also must be defined in outside of class.

int b;

public:

void set(int i,int j) { a=i; b=j;}

void show( );

};

int shared :: a;

void shared :: show( ) //a member function defined outside of the class body.

{

cout <<"This is static a: "<< a;

cout<<"\nThis is non_static b: " << b;

cout << "\n";

}

int main( )

{

shared x, y;

x.set(1, 1); //set a to 1

x.show( );

y.set(2, 2); //change a to 2

y.show( );

x.show( ); /* Here, a has been changed for both x and y

because a is shared by both objects.*/

return 0;  }
```

//an Example for static member function

```
#include<iostream.h>
class cl {
static int resource; //defining static data member.
public:
static int get_resource( ); //defining static member function.
void free_resource( ) {resource = 0;}
};
int cl :: resource;
int cl:: get_resource( )
{
if(resource) return 0 ; // resource already in use
else {
resource = 1;
return 1; //resource allocated to this object
}
}
int main( )
{
cl ob1, ob2;
/* get_resource( ) is static so may be called independent of any object.*/
if( cl :: get_resource( )) cout << "ob1 has resource\n ";
if( cl :: get_resource( )) cout << "ob2 denied resource\n ";
ob1.free_resource( );
if(ob2.get_resource( )) // can still call using object syntax
cout<<" ob2 can now use resource\n ";
return 0;
}
```

Notes//the static member function can't sign to any member function that is not static, also static member function can't be const.

```
//operator overloading (Completed Example)

#include <iostream.h>

class calc{

private:

int a,b,c;

public:

calc(){ a=0;b=0;c=0;}

void enter(){ cin>>this->a>>this->b>>this->c;}

void disp(){cout<<"a="<<a<<endl<<"b="<<b<<endl<<"c="<<c<<endl;}

calc operator +(calc obj) //the argument  is object of class and return type should be our class

{

calc temp;

temp.a=a+obj.a;

temp.b=b+obj.b;

temp.c=c+obj.c;

return temp;

}

calc operator -(calc obj)

{ calc temp;

temp.a=a-obj.a;

temp.b=b-obj.b;

temp.c=c-obj.c;

return temp;

}

calc operator /(calc obj)

{ calc temp;

//there is need to use some if else to avoid from dividing by 0;

temp.a=a/obj.a;

temp.b=b/obj.b;

temp.c=c/obj.c;

return temp;

}
```

```
calc operator *(calc obj)
{ calc temp;
temp.a=a*obj.a;
temp.b=b*obj.b;
temp.c=c*obj.c;
return temp;
}
calc operator %(calc obj)
{ calc temp;
temp.a=a%obj.a;
temp.b=b%obj.b;
temp.c=c%obj.c;
return temp;
}
};
main()
{
calc obj1 ,obj2,obj3;
obj1.enter() ; //try obj1.enter I  mean without ()
obj2.enter ();
obj3=obj1%obj2;
obj3.disp ();
}
```

Note// every time when we creat an object the constructor will be called automatically.

```
//Static data member & an example of overloading function.
#include <iostream.h>
class loc {
int num1, num2;
public:  static int a;
loc( ){a=6;}
loc(int a, int b) {
```

```
num1 = a;
num2 =b;
}
void show( ) {// a++;cout<<"a="<<a<<endl;
cout << num1<<" " ;
cout<< num2<< "\n ";
}
friend loc operator+ (int op1, loc op2) ;
};
int loc::a;
loc operator+ (int op1, loc op2)
{
loc temp;
temp.num1 =op1 + op2.num1;
temp.num2 =op1 + op2.num2;
return temp;
};
int main()
{ cout<<"a="<<loc.a<<endl;
loc ob1(10, 20), ob2(1,1) , ob3(7, 14);
ob1.show( ) ;
cout<<"a="<<loc.a<<endl;
return 0;
}
```

//an Example about static member function & showing you that it can only access to static data members.

```
#include <iostream.h>
class loc {
int num1, num2;
public:  static int a;
loc(int ){a=2;}
loc(int q, int b) {a=6;
```

```
num1 =q ;
num2 =b;
}
//Static member function can only access the static data members.
Static void show( ) { // a++;  cout<<"a="<<a<<endl;
//cout << num1<<" ";
//cout<< num2<< "\n ";   //cannot access num1 and num2
a=8; //it can access and make changes to static data members.
cout<<"static a="<<a<<endl; //can access
}
};
int loc::a;
int main()
{ cout<<"a="<<loc.a<<endl;
loc ob1(10, 20);
//ob1.show();
cout<<"2 argument constructor a= "<<ob1.a<<endl;
cout<<"outside accessing a="<<loc.a<<endl;
loc obj(7);
cout<<"defult constructor a="<<obj.a<<endl;
return 0;
}
```

```
//an Example about Friend Function.
#include<iostream.h>
class myclass {
int a, b;
friend int sum(myclass x);
public:
friend int sum(myclass x);//friend can be defined in public or in private
void set_ab(int i,int j);
};
void myclass :: set_ab(int i, int j)
```

```
{ a = i;b =j; }
// Note: sum( ) is not a member function of any class but it can access to data members of the class because it's friend of the class.
int sum(myclass x)
{
/* Because sum( ) is a friend of myclass, it can directly access a and b. */
return x.a + x.b;
}
int main( )
{
myclass n;
n.set_ab(3, 4);
cout<<sum(n)<<endl;
return 0;
}
```

```
//An example about Friend Class.
#include<iostream.h>
class TwoValues {
int a;
int b;
public:
TwoValues(int i, int j) {a = i, b= j;}
friend class Min;
};
class Min {
public:
int min(TwoValues x);
};
int Min::min (TwoValues x)
{return x.a < x.b? x.a: x.b;}
int main( )
{
```

```
TwoValues ob(10, 20);

Min m;

cout<< m.min(ob)<<endl;

return 0;

}
```

```
//Note about constructor .

#include<iostream.h>

class TwoValues {

int a;

int b;

public:

TwoValues(int i, int j)

{a = i, b= j; cout<<"Defult constructor of Towvalues called"<<endl;}

friend class Min;

};

class Min

 {

//Min(){cout<<"Defult constructor called...!"<<endl;} //important!  //we can't put //it like this in there!

//because when the declaration of objects we can't access to constructor

public:

Min(){cout<<"Defult constructor called...!"<<endl;}

int min(TwoValues x);

};

int Min::min (TwoValues x)

{return x.a< x.b? x.a: x.b;}

int main( )

{

TwoValues ob(10, 20),ob1(3,3);

Min m,n;

cout<< m.min(ob)<<endl;

return 0;
```

```
}
---------------------------
//Operator overloading.
#include <iostream.h>
class loc
{
int longitude, latitude;
public:
loc() { }
loc(int lg, int lt)
{
longitude = lg;
latitude = lt;
}
void show( ) {cout << longitude <<"  ";
cout<< latitude<< "\n ";
}
loc operator+ (loc op2);
loc operator++();
};
//Continued
//Overload +for loc.
loc loc::operator+ (loc op2)
{
loc temp;
cout<<"op.longitude= "<<op2.longitude<<endl;
cout<<"longitude= "<<longitude<<endl;
temp.longitude = op2.longitude+ longitude;
temp.latitude = op2.latitude+ latitude;
return temp;
};
loc loc:: operator++()
{
```

```
longitude++;
latitude++;
return *this ;
}
int main()
{
loc ob1(10, 20), ob2(5,30);
ob1.show( );
ob2.show( );
ob1= ob1 + ob2; // this mean we sent ob2 to our function   operator+ (ob2)...
ob1.show( ) ;
ob1++;
ob1.show();
return 0;
}
```

T.B//because friend function is not a member in class, that it cannot use "this" pointer.

Q/Why the friend function cannot use the "This" pointer in c++?

//Operator Overloading functions by using Friend functions.

```
#include <iostream.h>
class loc {
int num1, num2;
public:
loc( ){ }
loc(int a, int b) {
num1 = a;
num2 =b;
}
void show( ) {
cout << num1<<" " ;
cout<< num2<< "\n; ";
}
```

```
friend loc operator+ (loc op1, loc op2);

friend loc operator+ (int op1, loc op2) ;

};

loc operator+ (loc op1, loc op2)

{

loc temp;

temp.num1 = op1.num1 + op2.num1 ;

temp.num2 = op1.num2 + op2.num2 ;

return temp;

};

loc operator+ (int op1, loc op2)

{

loc temp;

temp.num1 =op1 + op2.num1;

temp.num2 =op1 + op2.num2;

return temp;

};

int main()

{

loc ob1(10, 20), ob2(5,30) , ob3(7, 14);

ob1.show( ) ;

ob2.show( );

ob3.show( );

//ob1= ob2 + 10; //both of these

ob3=10 + ob2; // are valid   //the function name::fname()   can't do that ,only //friend function can do it.

ob1.show( );        //because operator+() called by object that in left hand side

ob3.show( );        //  and 3+ is not an object that it cannot call the function,  for //this reason we used the friend function.

return 0;

}
```

Note// Binary operator like (+=) takes only one argument!

```
//I-Inheritance.

#include <iostream.h>

class base {

protected:

int i ,j ; //private to base , but accessible by derived because it's protected

public:

void set ( int a , int b) { i= a; j= b; }

void show( ) { cout<<i << " " << j << "\n"; }

private:

int q;

void     func(int a){q=a;cout<<"q="<<q<<endl;}

};

class derived : public base {

int k;

public:

// derived may access base's i and j by member functions

void setk( ) {k=i*j ;}

void showk( ) { cout <<k << " \n " ;}

void showme(){cout<<"i="<<i<<endl;} //as you can see we can access to protected
//member data.

};

int main( )

{

derived ob;

//ob.func (4);  //Cannot Access because fun() is a function that is defined in //private part
of the base class.

ob.set(2, 3) ; // ok, known to derived

ob.show( ) ; // ok, known to derived

ob.setk( );

ob.showk( );

return 0;  }
```

```
//II-Inheritance This program won't compile.
#include<iostream.h>
class base {
int i , j;
public:
void set( int a , int b) { i= a; j= b; }
void show( ) { cout<<i << " " << j << " \n "; }
protected:
int ca;
void ss(int x){ca=x; cout<<"calldee....!<<endl";}
};

// Public elements of base class are private in derived class.
class derived : private base {
int k;
public:
derived(int x){ k=x; }
void showk( ) { cout << k << " \n " ; }
};
int main( )
{
derived ob(3);
//ob.ss (4); //error ,can't access to protected
//ob.set(1 ,2); // error, can.t access set( )
//ob.show( ); // error, can.t access show( )
//because show() & set() are functions that defined in public part of the base //class and
ss() is a function that is defined in protected part.
return 0;
}
```

//Remaining the Destructing and Constructing (which of them called first).

```
#include <iostream.h>
class base {
public:
base ( ) { cout << "Constructing base \n";}
~ base( ) { cout << "Destructing base\n" ; }
};
class derived : public base {
public:
derived( ) { cout <<"Constructing derived\n" ; }
~derived( ) { cout<< "Destructing derived\n" ; }
};
int main ( )
{
derived ob;
// do nothing but construct and destruct ob
return 0;
}
```

//III-Inheritance and their constructor and destructor (which of them called first).

```
#include <iostream.h>
class base {
public:
//base ( ) { cout << "Constructing base \n";}
~ base( ) { cout << "Destructing base\n" ; }
protected:
base(){cout<<"Constructing protected Base \n";}
//      ~base(){cout<<"Constructing protected Base \n";}
};
class derived : public base {
public:
derived( ) { cout <<"Constructing derived\n" ; }
~derived( ) { cout<< "Destructing derived\n" ; }
```

```
};
int main ( )
{
derived ob;
// do nothing but construct and destruct ob
return 0;
}
```

//V-Inheritance and their constructor and destructor (which of them called first).

```
#include<iostream.h>
class base {
public:
base ( ){ cout<<" Constructing base \n ";}
~base( ){ cout<<" Destructing base\n " ; }
};
class derived1 : public base {
public:
derived1  ( ){ cout<<" Constructing derived1\n " ; }
~derived1 ( ){ cout<<" Destructing derived1\n " ; }
};
class derived2 : public derived1 {
public:
derived2 ( ){ cout<<" Constructing derived2\n " ; }
~derived2( ){ cout<< " Destructing derived2\n " ; }
};
int main ( )
{
derived2 ob;
// construct and destruct ob
return 0;
}
```

T.B// The base and Derived Constructor can be different argument

----

//Virtual class vC++ 2008.

```
#include "stdafx.h"
#include"iostream"
using namespace System;

class base
{
public:
void show( ) { std::cout<< "Base Show"<< "\n"; }
virtual void print(int x) { std::cout <<" Base Printed = " << x;}
};
class derived : public base
{
public:
void print(int x){  std::cout <<" Derived Printed = " << x;}
};
int main(array<System::String ^> ^args)
{
int a,b,c;

base *k; //now the base class cannot create objects because its abstract class , abstract class that using virtual functions.
derived ob;
k=&ob;
k->print (4); //We called the derived print...!

std::cin>>a;
return 0;
}
```

```
// cpp.cpp : main project file.

#include "stdafx.h"

#include"iostream"

using namespace System;

class base

{

public:

void show( ) { std::cout<< "Base Show"<< "\n"; }

virtual void print(int x) { std::cout <<" Base Printed = " << x;}

};

class derived : public base

{  public:

void print(int x){  std::cout <<" Derived Printed = " << x;}

};

int main(array<System::String ^> ^args)

{

int a,b,c;

base *k; //now the base class cannot create class because it's an abstract class , abstract class is that using the virtual functions.

derived ob;

k=&ob;

k->print (4); //We called the derived print...!

std::cin>>a;

return 0; }
```

T.B// the Virtual function :

1- Cannot be friend function.

2- Cannot be static member.

3- Cannot be constructor.

To be continue...